CLAIMS:

Having thus described our invention, what we claim as new, and desire to secure by Letters Patent is:

1. A system for secure data storage and retrieval comprising:

a storage device for storing encrypted data;

means at a client device for encrypting data prior to writing data blocks at said storage device, said encrypting means employing encryption capable of protecting individual data blocks against modification, relocation and replay for each data block written to said storage device;

means for generating an integrity value corresponding to one or more data blocks, said integrity value comprising information for preventing modification of data for each data block written to said storage device;

means for storing said integrity values of written data blocks;

means at said client device for decrypting said encrypted data accessed from said storage device; and,

means for performing an integrity check at said client device utilizing stored integrity values corresponding to stored data blocks being accessed, wherein said integrity check protects the integrity of contents stored in said storage device.

2. The system as claimed in Claim 1, wherein said encryption means generates encrypted cipher text data blocks that are a function of plaintext data included in said data block and a first encryption key.

3. The system as claimed in Claim 2, wherein said encryption means implements a whitening value which is a function of a second encryption key, an address location for said storage block, and a version number indicating a block write increment, said encryption means further generating cipher text data blocks that are additionally a function of said whitening value.

4. The system as claimed in Claim 2, wherein said encryption means employs an algorithm including one selected from DES or AES encryption schemes.

5. The system as claimed in Claim 3, wherein said means for storing said integrity values of written data blocks further includes means for generating an integrity tree structure, said integrity tree structure storing integrity values corresponding to each disk block written to said storage device.

6. The system as claimed in Claim 5, wherein said integrity tree comprises a hierarchical data structure, said hierarchical data structure including two or more layers of integrity data structures, each successive layer of integrity data structures including meta-data protecting integrity of data at an immediate prior layer.

7. The system as claimed in Claim 6, wherein said hierarchical data structure includes said written encrypted data blocks at a first layer, and a succeeding layer of meta-data blocks, each meta-data block including data structures representing a plurality of disk blocks written at said first layer, each meta-data block data structure comprising an integrity value and a version number pair for each of said plurality of disk blocks.

8. The system as claimed in Claim 7, wherein said integrity tree includes a succeeding layer of higher level meta-data blocks for protecting a layer of meta-data blocks below, each higher level meta-data block comprising data structures representing a plurality of meta-data blocks, each higher level meta-data block data structure comprising an integrity value and version number pair generated for each of said plurality of meta-data blocks.

9. The system as claimed in Claim 6, wherein a top layer of said hierarchical data structure includes a root data structure for protecting integrity of all content written to said storage device.

10. The system as claimed in Claim 9, further comprising means for writing a data block to said storage device, said writing comprising means for updating a written data block's version number and checksum in the associated meta-data blocks, wherein updates to

checksum and version number values are performed at each successive meta-data layer corresponding to said written data block, including updating performed at said root data structure.

11. The system as claimed in Claim 9, wherein said means for performing an integrity check comprises means comparing integrity of data blocks to be read on a path from said root data structure via successive higher meta-data blocks and meta-data block layers until a desired data block at a first layer is read.

12. The system as claimed in Claim 1, wherein said storage device comprises a non-volatile or volatile storage device.

13. The system as claimed in Claim 1, wherein said storage device is remotely located from said client device, said encrypted blocks being written across a network link.

14. A method for secure data storage and retrieval comprising the steps of:

a) encrypting data to be written from a client device to a storage device for storing encrypted data, said encrypting utilizing an encryption scheme capable of protecting individual data blocks against modification, relocation and replay for each data block written to said storage device;

b) generating an integrity value corresponding to one or more written data blocks, said integrity value comprising information for preventing modification of data for each data block written to said storage device;

c) storing said integrity values of written data blocks;

d) decrypting the encrypted data accessed from said storage device; and,

e) performing an integrity check utilizing said stored integrity values corresponding to stored data blocks being accessed, said integrity check protecting the integrity of contents stored in said storage.

15. The method as claimed in Claim 14, wherein said encrypting data step a) includes generating encrypted cipher text data blocks that are a function of plaintext data included in said data block and a first encryption key.

16. The method as claimed in Claim 15, wherein said encrypting data step a) further includes generating a whitening value as a function of a second encryption key, an address location for said storage block, and a version number indicating a block write, and the generation of cipher text data blocks that are a function of said whitening value.

17. The method as claimed in Claim 15, wherein said encrypting step a) further employs an algorithm including one selected from DES or AES encryption schemes.

18. The method as claimed in Claim 14, wherein said storing step c) further includes the step of: generating an integrity tree structure for storing integrity values corresponding to each disk block written to said storage device.

19. The method as claimed in Claim 18, wherein said integrity tree structure comprises a hierarchical data structure, said hierarchical data structure including two or more layers of integrity data structures, each successive layer of integrity data structures including meta-data protecting integrity of data at an immediate prior layer.

20. The method as claimed in Claim 19, further comprising the step of: writing encrypted data blocks at a first layer of said hierarchical data structure, and writing a succeeding layer of meta-data blocks, each meta-data block including data structures representing a plurality of disk blocks written at said first layer, each meta-data block data structure comprising an integrity value and a version number pair for each of said plurality of disk blocks.

21. The method as claimed in Claim 20, further comprising the step of: writing a succeeding layer of higher level meta-data blocks for protecting a layer of meta-data blocks below, each higher level meta-data block comprising data structures representing a plurality of meta-data blocks, each higher level meta-data block data structure comprising an integrity value and version number pair for each of said plurality of meta-data blocks.

22. The method as claimed in Claim 21, further comprising the step of: generating a root data structure at a top layer of said hierarchical data structure for protecting integrity of all content written to said storage device.

23. The method as claimed in Claim 22, further comprising the steps of: writing a data block to said storage device, said writing including updating a written data block's version number and checksum in the associated meta-data blocks, and, said checksum and version number value updating being performed at each successive meta-data layer corresponding to said written data block, including updating performed at said root data structure.

24. The method as claimed in Claim 22, further comprising the step of: reading a data block from said storage device, said step e) of performing an integrity check including comparing integrity of data blocks to be read on a path from said root data structure via successive meta-data block layers until a desired data block is read from said first layer of said hierarchical data structure.

25. A program storage device readable by a machine, tangibly embodying a program of instructions executable by the machine to perform method steps for securely storing and accessing data, said method steps comprising the steps of:

a) encrypting data to be written from a client device to a storage device for storing encrypted data, said encrypting utilizing an encryption scheme capable of protecting individual data blocks against modification, relocation and replay for each data block written to said storage device;

b) generating an integrity value corresponding to one or more written data blocks, said integrity value comprising information for preventing modification of data for each data block written to said storage device;

c) storing said integrity values of written data blocks;

d) decrypting the encrypted data accessed from said storage device; and,

e) performing an integrity check utilizing said stored integrity values corresponding to stored data blocks being accessed, said integrity check protecting the integrity of contents stored in said storage device.

26. The program storage device readable by a machine as claimed in Claim 25, wherein said encrypting data step a) includes generating encrypted cipher text data blocks that are a function of plaintext data included in said data block and a first encryption key.

27. The program storage device readable by a machine as claimed in Claim 25, wherein said encrypting data step a) further includes generating a whitening value as a function of a second encryption key, an address location for said storage block, and a version number indicating a block write increment, said encrypting step generating cipher text data blocks that are additionally a function of said whitening value.

28. The program storage device readable by a machine as claimed in Claim 27, wherein said storing step c) further includes the step of: generating an integrity tree structure for storing integrity values corresponding to each disk block written to said storage device, said integrity tree structure comprising a hierarchical data structure including two or more layers of integrity data structures, each successive layer of integrity data structures including meta-data protecting integrity of data at an immediate prior layer.

29. The program storage device readable by a machine as claimed in Claim 28, further comprising the step of: writing encrypted data blocks at a first layer of said hierarchical data structure, and writing a succeeding layer of meta-data blocks, each meta-data block including data structures representing a plurality of disk blocks written at said first layer, each meta-data block data structure comprising an integrity value and a version number pair for each of said plurality of disk blocks.

30. The program storage device readable by a machine as claimed in Claim 29, further comprising the steps of: writing a succeeding layer of higher level meta-data blocks for protecting a layer of meta-data blocks below, each higher level meta-data block comprising data structures representing a plurality of meta-data blocks, each higher level meta-data block data structure comprising an integrity value and version number pair for each of said plurality of meta-data blocks; and, generating a root data structure at a top layer of said hierarchical data structure for protecting integrity of all content written to said storage device.

31. The program storage device readable by a machine as claimed in Claim 30, further comprising the steps of: writing a data block to said storage device, said writing including updating a written data block's version number and checksum in the associated meta-data

blocks, and, said checksum and version number value updating being performed at each successive meta-data layer corresponding to said written data block, including updating performed at said root data structure.

32. The program storage device readable by a machine as claimed in Claim 30, further comprising the step of: reading a data block from said storage device, said step e) of performing an integrity check including comparing integrity of data blocks to be read on a path from said root data structure via successive meta-data block layers until a desired data block is read from said first layer of said hierarchical data structure.